CLAIMS

1. A method for optimizing disk space and information access of a computer storage system, comprising:

receiving formatting parameters for a hard partition of said computer storage system;

formatting said hard partition into a plurality of soft partitions; and

building system structures based upon said formatting parameters, said system structures comprising:

a soft partition table and corresponding soft partition bit maps operable for specifying whether sectors on a disk are free or occupied, said soft partition table storing information for soft partitions;

a sector descriptor file storing sector descriptors, said sector descriptors comprising a series of encoded bytes operable for tracking physical sectors reserved for a file to hold file data; and

an offset entry file operable for specifying availability of storage blocks in said sector descriptor file.

2. The method of claim 1, further comprising:

setting customizable sector allocations for a file; and

setting conditions for sector allocation for a file.

3. The method of claim 1, wherein said formatting parameters include:

at least one hard partition size;

a drive name;

a sector size;

a sector descriptor storage block size; and

a soft partition size.

4. The method of claim 1, wherein said formatting parameters are selectable by a user.

5. The method of claim 1, further comprising:

periodically scanning said offset entry file for free entries;

searching for sector descriptors capable of being moved from an end point of said sector descriptor file; and

if found, moving said sector descriptors into said free entries.

6. The method of claim 1, wherein said sector descriptor file comprises a variable number of storage blocks dependent upon a number and size of sector descriptors occupying said sector descriptor file; and wherein further:

each of said variable number of storage blocks are equal in size; and

each bit in said offset entry file corresponds to a single storage block.

7. The method of claim 1, wherein sector descriptors of small size are stored in a file descriptor block, wherein a bit in said file descriptor block indicates whether said sector descriptor is stored internally or in said sector descriptor file.

8. The method of claim 1, wherein building a soft partition table further comprises:

assigning a group count for each entry of said soft partition table as null operable for indicating a non-assignment;

selectively indicating a read/write-protect for said each entry;

providing a starting location of said each entry's soft partition bit map via a pointer column;

establishing group counts of variable sizes to a first four entries in said soft partition table, if available, wherein each bit in said corresponding soft partition bit map refers to a single group count; and wherein further, said group counts indicate a number of contiguous sectors that each bit in said soft partition bit map represents;

assigning group counts to remaining entries as needed; and

reassigning new group counts to entries when all bits associated with an entry become zero.

9. The method of claim 8, wherein said group count refers to a flexible multiple of two assigned to said each entry;

10. The method of claim 1, wherein said sector descriptors comprise:

at least one group byte operable for setting a 512-gigabyte boundary for a file;

at least one extended cylinder byte operable for setting an upper eight bits of cylinder/head characteristics currently pointed to by a sector descriptor;

at least one middle cylinder byte operable for setting a middle eight bits of said cylinder/head characteristics currently pointed to by a sector descriptor;

at least one cylinder/head byte operable for setting a lower eight bits of cylinder/head characteristics currently pointed to by a sector descriptor; and

at least one sector start byte operable for setting a current starting sector in a cylinder/head location currently pointed to by a sector descriptor, said sector start byte including two bits operable for determining which of three preceding bytes will change, if any, and whether a sector count byte follows;

wherein said sector count byte sets a number of contiguous multiples of an MSC pointed to by a current location in a sector descriptor.

11. The method of claim 2, wherein said setting customizable sector allocations for a file further includes:

assigning a minimum sector count (MSC) to a file, said MSC comprising a minimum number of contiguous sectors allocatable to a file;

wherein said MSC is determined by file size and file type.

12. The method of claim 11, further comprising:

assigning a new MSC in response to a substantial alteration of said file size, comprising:

computing a number of sectors required for said altered file;

comparing a number of sectors required for said file prior to alteration, with said number of sectors required for said altered file;

purging a sector descriptor associated with said file;

setting a new MSC;

setting said number of sectors in accordance with said altered file;

setting a new sector descriptor for said altered file;

formatting said new sector descriptor; and

storing new settings for said altered file.

13. The method of claim 12, wherein said formatting said new sector descriptor comprises:

searching a soft partition bit map for continuous groups of sectors with counts capable of accommodating said MSC;

searching alternative soft partition bit maps for continuous groups of sectors when said soft partition bit map is full;

computing values of sector descriptor bytes in accordance with a first free group found; and

moving said bytes to a sector descriptor location (SDL) and temporary sector descriptor (TSD), wherein said first values in said TSD are equivalent to said values in said SDL;

wherein further said SDL refers to a location in memory where a next formatted byte for said sector descriptor is placed.

14. A storage medium encoded with machine-readable computer program code for optimizing disk space and information access of a computer storage system, the storage medium comprising instructions for causing a computer to implement a method comprising:

receiving formatting parameters for a hard partition of said computer storage system;

formatting said hard partition into a plurality of soft partitions; and

building system structures based upon said formatting parameters, said system structures comprising:

a soft partition table and corresponding soft partition bit maps operable for specifying whether sectors on a disk are free or occupied, said soft partition table storing information for soft partitions;

a sector descriptor file storing sector descriptors, said sector descriptors comprising a series of encoded bytes operable for tracking physical sectors reserved for a file to hold file data; and

an offset entry file operable for specifying availability of storage blocks in said sector descriptor file.

15. The storage medium of claim 14, further comprising instructions for causing said computer to implement a method comprising:

setting customizable sector allocations for a file; and

setting conditions for sector allocation for a file.

16. The storage medium of claim 14, wherein said formatting parameters include:

at least one hard partition size;

a drive name;

a sector size;

a sector descriptor storage block size; and

a soft partition size.

17. The storage medium of claim 14, wherein said formatting parameters are selectable by a user.

18. The storage medium of claim 14, further comprising instructions for causing said computer to implement a method comprising:

periodically scanning said offset entry file for free entries;

searching for sector descriptors capable of being moved from an end point of said sector descriptor file; and

if found, moving said sector descriptors into said free entries.

19. The storage medium of claim 14, wherein said sector descriptor file comprises a variable number of storage blocks dependent upon a number and size of sector descriptors occupying said sector descriptor file; and wherein further:

each of said variable number of storage blocks are equal in size; and

each bit in said offset entry file corresponds to a single storage block.

20. The storage medium of claim 14, wherein sector descriptors of small size are stored in a file descriptor block, wherein a bit in said file descriptor block indicates whether said sector descriptor is stored internally or in said sector descriptor file.

21. The storage medium of claim 14, wherein building a soft partition table further comprises instructions for causing said computer to implement a method comprising:

assigning a group count for each entry of said soft partition table as null operable for indicating a non-assignment;

selectively indicating a read/write-protect for said each entry;

providing a starting location of said each entry's soft partition bit map via a pointer column;

establishing group counts of variable sizes to a first four entries in said soft partition table, if available, wherein each bit in said corresponding soft partition bit map refers to a single group count; and wherein further, said group counts indicate a number of contiguous sectors that each bit in said soft partition bit map represents;

assigning group counts to remaining entries as needed; and

reassigning new group counts to entries when all bits associated with an entry become zero.

22. The storage medium of claim 21, wherein said group count refers to a flexible multiple of two assigned to said each entry.

23. The storage medium of claim 14, wherein said sector descriptors comprise:

at least one group byte operable for setting a 512-gigabyte boundary for a file;

at least one extended cylinder byte operable for setting an upper eight bits of cylinder/head characteristics currently pointed to by a sector descriptor;

at least one middle cylinder byte operable for setting a middle eight bits of said cylinder/head characteristics currently pointed to by a sector descriptor;

at least one cylinder/head byte operable for setting a lower eight bits of cylinder/head characteristics currently pointed to by a sector descriptor; and

at least one sector start byte operable for setting a current starting sector in a cylinder/head location currently pointed to by a sector descriptor, said sector start byte including two bits operable for determining which of three preceding bytes will change, if any, and whether a sector count byte follows; wherein said sector count byte sets a number of contiguous multiples of an MSC pointed to by a current location in a sector descriptor.

24. The storage medium of claim 15, wherein said setting customizable sector allocations for a file further includes instructions for causing said computer to implement a method comprising:

assigning a minimum sector count (MSC) to a file, said MSC comprising a minimum number of contiguous sectors allocatable to a file;

wherein said MSC is determined by file size and file type.

25. The storage medium of claim 24, further comprising instructions for causing said computer to implement a method comprising:

assigning a new MSC in response to a substantial alteration of said file size, comprising:

computing a number of sectors required for said altered file;

comparing a number of sectors required for said file prior to alteration, with said number of sectors required for said altered file;

purging a sector descriptor associated with said file;

setting a new MSC;

setting said number of sectors in accordance with said altered file;

setting a new sector descriptor for said altered file;

formatting said new sector descriptor; and

storing new settings for said altered file.

26. The storage medium of claim 25, wherein said formatting said new sector descriptor comprises instructions for causing said computer to implement a method comprising:

searching a soft partition bit map for continuous groups of sectors with counts capable of accommodating said MSC;

searching alternative soft partition bit maps for continuous groups of sectors when said soft partition bit map is full;

computing values of sector descriptor bytes in accordance with a first free group found; and

moving said bytes to a sector descriptor location (SDL) and temporary sector descriptor (TSD); wherein said first values in said TSD are equivalent to said values in said SDL;

wherein further said SDL refers to a location in memory where a next formatted byte for said sector descriptor is placed.

27. A system for optimizing disk space and information access of a computer storage system via a disk management utility, comprising:

a file initialization component performing a method comprising:

receiving formatting parameters for a hard partition of said computer storage system;

formatting said hard partition into a plurality of soft partitions; and

building system structures based upon said formatting parameters, said system structures comprising:

a soft partition table and corresponding soft partition bit maps operable for specifying whether sectors on a disk are free or occupied, said soft partition table storing information for soft partitions;

a sector descriptor file storing sector descriptors, said sector descriptors comprising a series of encoded bytes operable for tracking physical sectors reserved for a file to hold file data; and

an offset entry file operable for specifying availability of storage blocks in said sector descriptor file.

28. The system of claim 27, further comprising:

a file allocation component, comprising:

a sector descriptor data stream;

a file descriptor block;

a file pointer table and file/sector descriptor table; and

a data management block;

wherein said file allocation component performs a method, comprising:

setting customizable sector allocations for a file; and

setting conditions for sector allocation for a file.

29. The system of claim 28, wherein said sector descriptor data stream comprises:

at least one group byte operable for setting a 512-gigabyte boundary for a file.

30. The system of claim 29, wherein said sector descriptor data stream further comprises:

at least one extended cylinder byte operable for setting an upper eight bits of cylinder/head characteristics currently pointed to by a sector descriptor.

31. The system of claim 30, wherein said sector descriptor data stream further comprises:

at least one middle cylinder byte operable for setting a middle eight bits of said cylinder/head characteristics currently pointed to by a sector descriptor.

32. The system of claim 31, wherein said sector descriptor data stream further comprises:

at least one cylinder/head byte operable for setting a lower eight bits of cylinder/head characteristics currently pointed to by a sector descriptor.

33. The system of claim 32, wherein said sector descriptor data stream further comprises:

at least one sector start byte operable for setting a current starting sector in a cylinder/head location currently pointed to by a sector descriptor, said sector start byte including two bits operable for determining which of three preceding bytes will change, if any, and whether a sector count byte follows; wherein said sector count byte sets a number of contiguous multiples of an MSC pointed to by a current location in a sector descriptor.

34. The system of claim 28, wherein said file descriptor block stores file entries of 64 bytes per entry.